



# The Highly Integrated Vehicle Ecosystem (HIVE): A Platform for Managing the Operations of On-Demand Vehicle Fleets

Robert Fitzgerald, Nicholas Reinicke, and Matthew Moniot

*National Renewable Energy Laboratory*

**NREL is a national laboratory of the U.S. Department of Energy  
Office of Energy Efficiency & Renewable Energy  
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at [www.nrel.gov/publications](http://www.nrel.gov/publications).

Contract No. DE-AC36-08GO28308

**Technical Report**  
NREL/TP-5400-80682  
September 2021



# The Highly Integrated Vehicle Ecosystem (HIVE): A Platform for Managing the Operations of On-Demand Vehicle Fleets

Robert Fitzgerald, Nicholas Reinicke, and Matthew Moniot

*National Renewable Energy Laboratory*

## **Suggested Citation**

Fitzgerald, Robert, Nicholas Reinicke, and Matthew Moniot. 2021. *The Highly Integrated Vehicle Ecosystem (HIVE): A Platform for Managing the Operations of On-Demand Vehicle Fleets*. Golden, CO: National Renewable Energy Laboratory. NREL/TP-5400-80682. <https://www.nrel.gov/docs/fy21osti/80682.pdf>.

**NREL is a national laboratory of the U.S. Department of Energy  
Office of Energy Efficiency & Renewable Energy  
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at [www.nrel.gov/publications](http://www.nrel.gov/publications).

Contract No. DE-AC36-08GO28308

**Technical Report**  
NREL/TP-5400-80682  
September 2021

National Renewable Energy Laboratory  
15013 Denver West Parkway  
Golden, CO 80401  
303-275-3000 • [www.nrel.gov](http://www.nrel.gov)

## NOTICE

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the National Renewable Energy Laboratory through the Autonomous Urbanization: Mobility and Communities Laboratory Directed Research and Development project. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government.

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at [www.nrel.gov/publications](http://www.nrel.gov/publications).

U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via [www.OSTI.gov](http://www.OSTI.gov).

*Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.*

NREL prints on paper that contains recycled content.

## Acknowledgments

The authors acknowledge the contributions of several others without whom HIVE would have never been created, including Clement Rames, Brennan Borlaug, Tom Grushka, Jacob Holden, and Eric Wood.

## List of Acronyms

BEV	battery electric vehicle
DOE	U.S. Department of Energy
FHV	for-hire vehicle
HIVE	Highly Integrated Vehicle Ecosystem
LDRD	Laboratory Directed Research and Development
MoD	mobility on demand
OSM	OpenStreetMap
PHEV	plug-in hybrid electric vehicle
SAEV	shared autonomous electric vehicle
YCB	Yellow Cab

## Abstract

This paper introduces the Highly Integrated Vehicle Ecosystem (HIVE), a transportation modeling tool developed by a research team within the Center for Integrated Mobility Sciences group at the National Renewable Energy Laboratory. HIVE is an agent-based supply/demand model for mobility on demand (MoD) that mixes agent-based modeling and centralized dispatch for automated and human-driven fleets and ride-hail passengers. Research questions using HIVE span multiple categories, including intelligent fleet planning (e.g., assessing fleet, battery, and infrastructure investment decisions), intelligent fleet control (e.g., charge management, vehicle dispatching), and strategic business model decision-making (e.g., depot-based full-time drivers versus gig-based drivers, human-driven versus automated). The components of the HIVE model are explained and then HIVE is demonstrated in a case study using demand data from the New York City Taxi & Limousine Commission data set.

Additional summary information describing how HIVE is used in support of NREL research can be found online: <https://www.nrel.gov/transportation/data-tools.html>. HIVE is used by transportation researchers as a standalone model and also informs assumptions within the EVI-OnDemand module in NREL's EVI-X infrastructure tool suite.

## Contents

<b>Disclaimer</b> . . . . .	<b>i</b>
<b>Acknowledgments</b> . . . . .	<b>ii</b>
<b>List of Acronyms</b> . . . . .	<b>iii</b>
<b>Abstract</b> . . . . .	<b>iv</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
<b>2 Model</b> . . . . .	<b>2</b>
2.1 Physics-Level Modeling . . . . .	2
2.1.1 Energy . . . . .	2
2.1.2 Space . . . . .	3
2.1.3 Time . . . . .	5
2.2 Asset-Level Modeling . . . . .	5
2.2.1 Vehicle . . . . .	5
2.2.2 Request . . . . .	7
2.2.3 Station . . . . .	7
2.2.4 Base . . . . .	8
2.2.5 Membership . . . . .	8
2.3 Execution-Level Modeling . . . . .	9
2.3.1 Simulation State . . . . .	9
2.3.2 Control Module . . . . .	10
2.3.3 Execution Context . . . . .	12
2.4 Model Inputs . . . . .	13
<b>3 Case Study</b> . . . . .	<b>14</b>
3.1 Case Study Inputs . . . . .	14
3.2 Results . . . . .	14
<b>4 Conclusion and Future Work</b> . . . . .	<b>17</b>
<b>References</b> . . . . .	<b>18</b>

## List of Figures

Figure 1. Architecture of the HIVE model . . . . .	2
Figure 2. Default energy consumption and charge tapering for the BEV type . . . . .	4
Figure 3. Conceptual representation of an EntityPosition . . . . .	4

Figure 4.	Finite-state machine of vehicle states in HIVE . . . . .	7
Figure 5.	Example of memberships in HIVE . . . . .	9
Figure 6.	HIVE update process . . . . .	9
Figure 7.	Vehicle update . . . . .	10
Figure 8.	Example of using H3 for a charging station search. The vehicle is seeking a charging location at one of a set of station alternatives, shown as drop points on the map. The search is conducted over an expanding ring of H3 hex grid cells at a higher level of aggregation. . . . .	11
Figure 9.	Example vehicle state update . . . . .	12
Figure 10.	Inputs to the HIVE model . . . . .	13
Figure 11.	Sample vehicle trajectory and state-of-charge evolution throughout a 24-hour simulation . . . . .	15
Figure 12.	Sample load profiles from stations in HIVE with varying membership access types (FHV exclusive, YCB exclusive, and shared) . . . . .	15
Figure 13.	Request service across YCB and FHV fleets exposed to distinct request populations . . . . .	16

## List of Tables

Table 1.	Vehicle Attributes . . . . .	6
Table 2.	Driver States and Their Shift Behaviors . . . . .	6
Table 3.	Vehicle States . . . . .	6
Table 4.	Attributes of a Request . . . . .	8
Table 5.	Attributes of a Station . . . . .	8
Table 6.	Attributes of a Base . . . . .	8

## 1 Introduction

The transportation industry is on the cusp of rapid transformation, with simultaneous trends of vehicle electrification, automation, and sharing representing a significant departure from the current paradigm of predominantly privately owned and operated gasoline vehicles. However, each trend poses unique opportunities and obstacles. Vehicle electrification can significantly reduce greenhouse gas and particulate emissions from the transportation sector [1] [2] and reduce operating costs [3], but successful electrification will require access to a robust network of charging infrastructure [4]. Vehicle automation can greatly reduce avoidable traffic injuries and fatalities [5], but with possible increases in the energy consumption at both the vehicle and fleet levels [6]. Vehicle sharing offers more convenient door-to-door transportation options and potentially reduces the need for private vehicle ownership [7], but it also increases vehicle-miles traveled through deadheading miles between productive trips [8].

Evaluated separately, there are opportunities and obstacles associated with each of the emerging trends of vehicle sharing, electrification, and automation. It is clear that attention will be needed toward mitigating potential negative side effects, such as minimal ride pooling, insufficient and poorly planned recharging infrastructure, and the prominence of zero-passenger miles during operation of these services. Shared autonomous electric vehicle (SAEV) fleets—which will be more broadly referred to as mobility-on-demand (MoD) services—is an anticipated transportation technology development that combines each of these three trends into a transportation service that could potentially increase transportation efficiency and mobility options while also featuring low financial and environmental costs.

The performance of such fleets has been well researched in recent years, primarily through agent-based models simulating automated fleets as accommodating passenger-trip demand. Early research assessed fleet capabilities to service trip demand using gasoline vehicles in defined operating areas assuming the presence of a centralized dispatcher [9, 10] and often without incorporating a real-world roadway network [11–13]. More recent research has built upon these earlier dispatching models by considering a shared fleet of electric vehicles servicing trips, rather than gasoline-powered vehicles [14–17]. Battery electric fleets are more challenging to operate compared to gasoline fleets due to range constraints and the necessity of a recharging network. Additional topics of inquiry found in the literature relate to strategic siting of MoD charging infrastructure [18] and the implications of fleets comprising human drivers [19]. Each of these research topics utilizes distinct simulation platforms with varying strengths and limitations [20].

The complexities of these emerging transportation trends motivated the creation of the Highly Integrated Vehicle Ecosystem (HIVE), a transportation modeling tool developed by researchers in the Center for Integrated Mobility Sciences group at the National Renewable Energy Laboratory. HIVE is a fixed-increment, discrete-event, agent-based ride-hail simulation written in Python. The tool was conceived to support a rapid end-to-end research process, with a minimal level of effort sufficient to run simulation and analysis of MoD scenarios. Research questions using HIVE span multiple categories, including intelligent fleet planning (e.g., assessing fleet, battery, and infrastructure investment decisions), intelligent fleet control (e.g., charge management, vehicle dispatching), and strategic business model decision-making (e.g., depot-based full-time drivers versus gig-based drivers, human-driven versus automated).

HIVE has been designed from the ground up to facilitate co-simulation, the use of high-performance computing resources, and data-driven control. By maintaining a loose coupling between the state of the simulation and the execution of the simulation physics, it is easy to inject additional modules and even to externally control execution of HIVE from a complementary model. By choosing immutable semantics on the state model, the state of the simulation more readily lends itself to message passing and rollback techniques used in distributed simulation. Additionally, these choices lead to simpler integration into popular learning environments such as OpenAI Gym [21] to explore artificial-intelligence-based control techniques such as deep reinforcement learning.

The objective of this technical report is to introduce HIVE and demonstrate its functionality. Section 2 provides an overview of HIVE’s structure, illuminating how pieces of the model function and interact with each other. The model is then applied in a sample study in Section 3. Finally, Section 4 concludes with a future perspective detailing upcoming model developments in support of ongoing and planned research.

## 2 Model

The HIVE model architecture is broken out into three distinct levels: execution, asset, and physics, as shown in Figure 1. The physics level consists of simulation representations of real-world concepts including energy, time, and space. The asset level consists of the building blocks that describe the various entities in the system including vehicles, requests, stations, and bases, as well as their relationship to one another. The execution level consists of the concepts and data structures that tie all of the entities together and rules to manage their interaction.

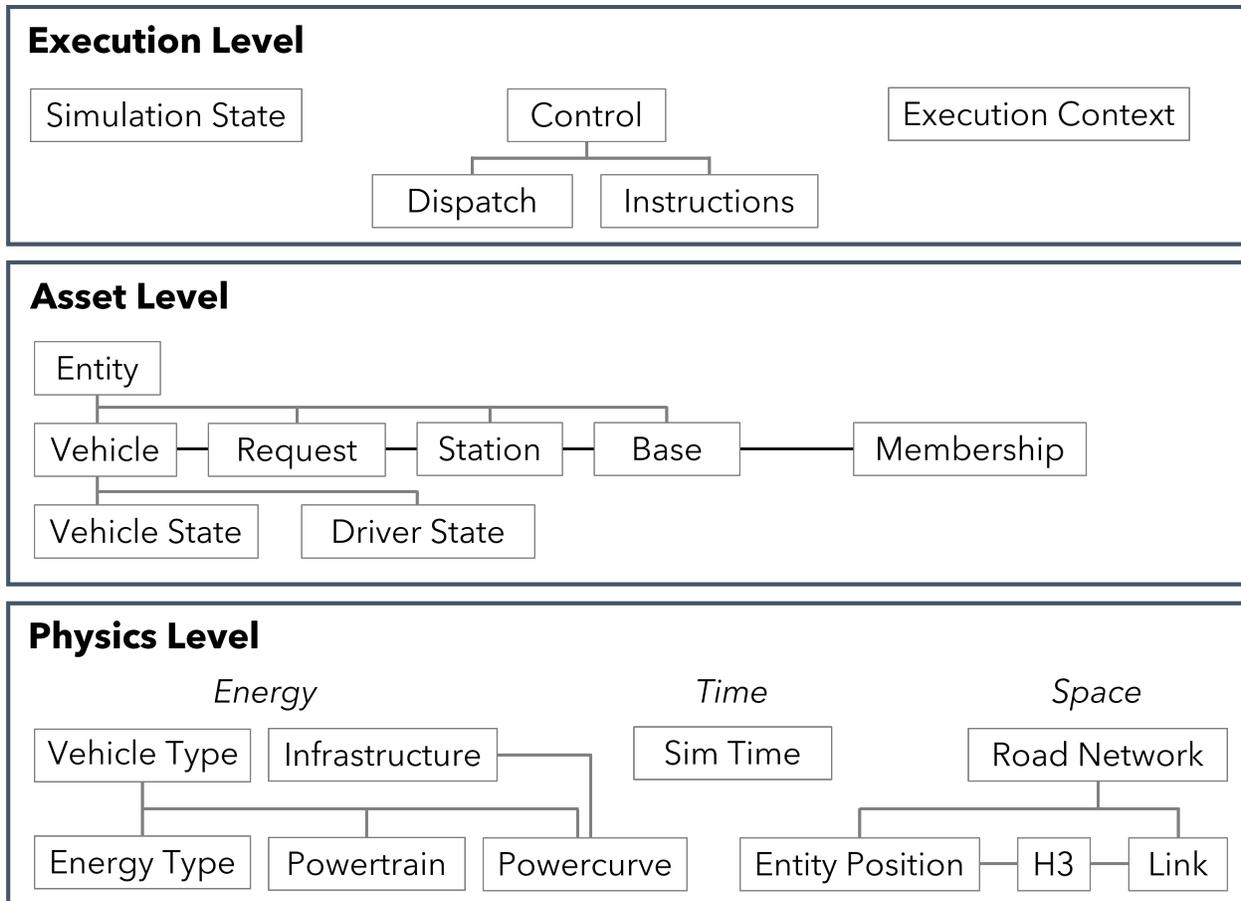


Figure 1. Architecture of the HIVE model

In the following sections, each layer of the simulation is summarized, followed by an overview of the corresponding file inputs for the model.

### 2.1 Physics-Level Modeling

#### 2.1.1 Energy

##### Vehicle Types

HIVE currently models two general vehicle types: battery electric vehicles (BEVs) and internal combustion engine vehicles. Each of these vehicle types utilizes a specific energy type: electricity (in units of kilowatt-hours) and gasoline (in units of gallons), respectively. In addition, the vehicle types model the unique vehicle dynamics for their respective vehicle class such as energy consumption and energy storage. Each general vehicle type can be configured to represent a specific vehicle model. For example, half of a vehicle fleet could be composed of BEVs

with a 50-kWh battery and a nominal energy consumption rate of 0.225 kWh per mile, whereas the other half could be composed of BEVs with a 25-kWh battery and a nominal energy consumption rate of 0.3 kWh per mile.

### Energy Storage and Addition

Energy storage in HIVE is modeled by a simple capacity model in which each vehicle tracks their respective energy level,  $e_{veh}$ , over the simulation in the range  $0 \leq e_{veh} < c_{veh}$ , where  $c_{veh}$  is the energy capacity parameterized by the vehicle type.

When adding energy to a vehicle, the simulation obeys a set of limits imposed by the rate of the charger and the vehicle type. In the case of the internal combustion engine vehicle type, energy is added at a constant linear rate as defined by the fuel source. For the BEV type, energy is added at the charger's constant power if the power is below a charging tapering threshold. Otherwise, if the charger power meets or exceeds this threshold, energy will be added at diminishing rate proportional to the state of charge of the vehicle as defined by Equation 2.1.

$$E^+ = \sum_{t=0}^T (p * t * r(e_{veh})) \quad (2.1)$$

where  $E^+$  is the energy added over the fueling period,  $T$ ;  $p$  is the incoming fueling rate; and  $r(e_{veh})$  is the rate-limiting factor as a function of vehicle energy level. See Figure 2a for the default rate-limiting curve,  $r(e_{veh})$ , for the BEV type. The rate-limiting curves are derived from observed charge tapering behavior in BEVs [22] and labeled as a Powercurve in the HIVE model.

### Energy Consumption

Energy consumption in HIVE is modeled with an internal powertrain object and specifies consumption at the level of a traversed link (see Section 3.1.2) using Equation 2.2.

$$E^- = \sum_l^L r(v_l) * d_l \quad (2.2)$$

where  $E^-$  is the total energy consumed in a time period,  $L$  is the set of all links traversed in the time period,  $r(v_l)$  is the energy consumption rate as a function of vehicle average velocity on link  $l$ , and  $d$  is the distance of link  $l$ .

The energy consumption rates have been derived from vehicles simulated in the Future Automotive Systems Technology Simulator (FASTSim) software [23]. Figure 2b shows the default energy consumption by speed for the BEV type.

HIVE also models energy consumption while a vehicle is in the idle vehicle state (engine is on but vehicle is stationary). This is computed as a constant energy consumption rate over the time that the vehicle is idling. The default idle energy consumption rate is 0.8 kWh/hour for the BEV type and 0.2 gallons/hour for the internal combustion engine vehicle type, but these values can be configured independently.

## 2.1.2 Space

### H3

Space is modeled in HIVE using the H3 hierarchical spatial indexing system [24], in which the atomic spatial unit is a hexagon with a fixed diameter (the default is a hexagon with a diameter of approximately 1 meter but can be configured to any H3 spatial resolution). This system of representing space allows for quick testing of spatial equality for various simulation entities. In addition, the hierarchical nature of the indexing allows for more efficient search algorithms. To this end, HIVE stores simulation entity IDs in two data stores: One data store indexes entities by the simulation H3 resolution, and the other indexes them at a coarser search H3 resolution.

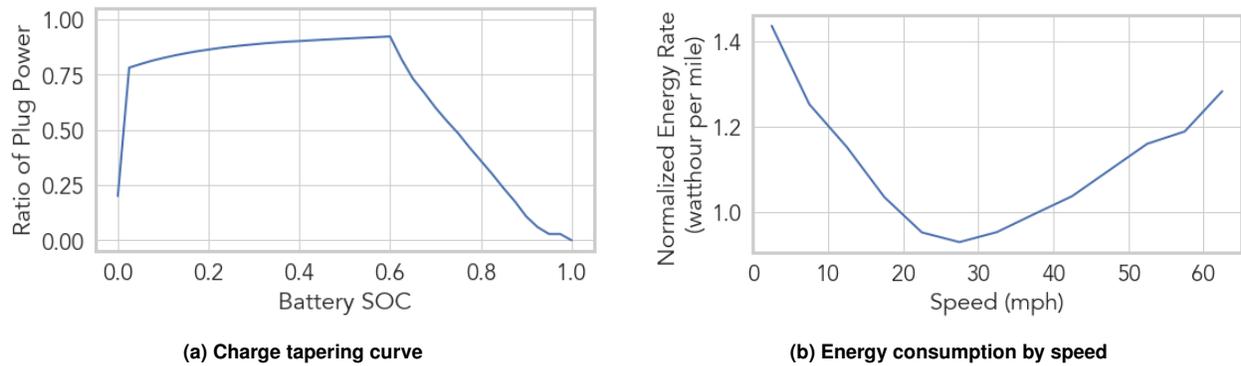


Figure 2. Default energy consumption and charge tapering for the BEV type

### Link

The next level up in HIVE’s spatial representation system is the link. The link represents a road as a directed edge in which the start and end nodes are represented as H3 hexagons. Each link stores relevant properties such as link distance (in kilometers) and link speed (in kilometers/hour), which are exogenous to the model.

### Entity Position

Combining the abstractions of the H3 geoid and the link, the instantaneous location of each entity is represented in HIVE using an EntityPosition object, which includes a link ID and an H3 geoid. The link provides context with respect to direction and the geoid represents the physical location of the entity. Thus, to compare the locations of two distinct entities, the test is a simple equality of both the link ID and the geoid. Figure 3 demonstrates the concept of an EntityPosition.

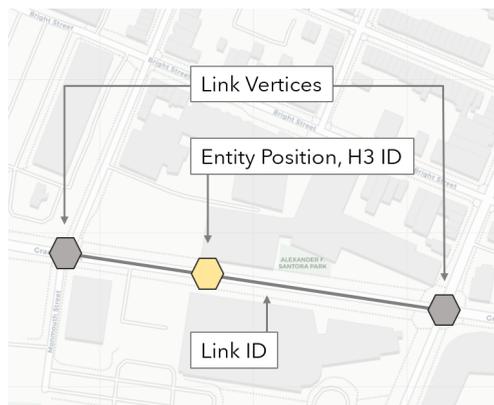


Figure 3. Conceptual representation of an EntityPosition

### Road Network

The final level in HIVE’s spatial representation system is the road network graph, which is composed of a collection of links. The model enforces that the graph must be strongly connected such that there is a path between all vertices in the graph, as well as planar—a constraint imposed by the H3 index.

There are currently two distinct types of road network graphs in the HIVE model: a pseudo-Euclidean graph and a graph derived from OpenStreetMap (OSM) data.

The pseudo-Euclidean graph is a complete digraph in which every pair of H3 hexagons is connected by a pair of unique links. This graph effectively allows the vehicles to travel in a straight line between any two hexagons in the

system. Links are generated dynamically when required so as not to incur a large memory overhead. All links in this road network are assigned a constant speed derived to represent a system-level average speed. This simplification of the road network representation is useful for running quick model tests.

The OSM graph uses links from the OSM data model sourced using the open-source OSMnx software library [25]. The links derive speed and distance properties from OSM. When introducing new entities into the road network with geospatial information (e.g., latitude and longitude), the entity location is snapped to the nearest link vertex.

### Routing

A route object is a collection of links that represents a vehicle's planned course of travel to a destination along the road network. During each time step, the vehicle traverses the route according to the link speed property. As the vehicle traverses over the links in the route, these are passed into the vehicle powertrain, which estimates energy consumption for the vehicle's movement.

Vehicle routing is computed within the scope of the road network. For the pseudo-Euclidean graph, all routes consist of a single link that directly connects the origin and destination. In the case of the OSM graph, routes are computed using Dijkstra's shortest path algorithm with travel time as the link weight.

### 2.1.3 Time

The atomic time unit in HIVE is a discrete time step that represents one or more seconds (configurable). Then, bounds for a simulation start time and end time are specified and the simulation is stepped over each time step that lies within these bounds. For example, in a 24-hour simulation with a time step size of 60 seconds, HIVE will be stepped 1,440 times.

## 2.2 Asset-Level Modeling

HIVE currently models four unique types of entities: vehicles, requests, stations, and bases. These four entity types interact in an agent-based model setting.

### 2.2.1 Vehicle

The vehicle in HIVE is the primary type of agent in the HIVE agent-based model. Internally, the vehicle state is modeled by a pair of disjoint finite-state machine models that represent behaviors of the driver and vehicle. As a result, all agents in HIVE will have a driver state and a vehicle state. All persistent vehicle and driver attributes are also stored on the vehicle. In the following sections, the persistent attributes of a vehicle are discussed, followed by subsections on the driver and vehicle finite-state machines.

#### Vehicle Attributes

The persistent variables of a vehicle include *fixed*, *stateful*, and *accumulative* fields. Fixed attributes include the vehicle's unique identifier, the number of seats available for passengers, and the memberships the vehicle supports (none, by default). These are understood to be held constant through a HIVE simulation. Stateful attributes include the vehicle state, the driver state, vehicle energy, and its current link on the road network. Each time step, these attributes govern the behavior of the agent and may be updated based on the vehicle's interaction. The default vehicle state is an idle state, in which a vehicle is on but immobile. The default driver state, `AutonomousAvailable`, is a driver that can always respond to instruction. Over the lifetime of the vehicle, the currency balance and distance traveled are accumulated in vehicle attributes. Currency balance incorporates all service payments from requests (revenue), as well as all operating costs (station charges). The complete set of vehicle attributes is listed in Table 1.

#### Driver State

The driver state of a HIVE agent is a simple model responsible for making the personal choices of an agent. Table 2 shows the full list of possible states for HIVE driver agents, which may be autonomous or human-driven. Autonomous drivers will accept all instructions from dispatch, and will return to base after some period of inactivity.

Attribute	Life Cycle	Type	Default
id	fixed	Id (String)	
total_seats	fixed	int	
membership	fixed	Membership	∅
vehicle_state	stateful	VehicleState	Idle
driver_state	stateful	DriverState	AutonomousAvailable
vehicle_energy	stateful	Map[EnergyType, float]	
position	stateful	EntityPosition	
currency	accumulative	float	0.0
distance_traveled_km	accumulative	float	0.0

**Table 1. Vehicle Attributes**

A human driver will only work within some provided time schedule and will return to their home at the end of their shift. To instantiate human drivers, additional parameters must be provided. A `home_base_id` (a valid base ID) signifies that a driver has a home, where they may have private charging capabilities. A `schedule_id` references a corresponding work schedule that describes the time range when the driver is willing to respond to dispatch instruction.

Driver State	Shift Behavior
Autonomous	Always available to work
Human Available	On-shift driver
Human Unavailable	Off-shift driver

**Table 2. Driver States and Their Shift Behaviors**

#### Vehicle State

The vehicle state model describes the intent or task applied to an agent at a given time. The complete list of vehicle states is listed in Table 3. Beginning from the default idle state, a driver will transition to various vehicle states throughout the simulation as their role in the fleet and circumstances change.

Vehicle State	Moving?	Description
Idle	No	Vehicle engine is on, but not performing any task
Repositioning	Yes	Deadheading without servicing any request
DispatchBase	Yes	Driving to a base
ReserveBase	No	Vehicle engine is off, parked at base/home
ChargingBase	No	Plugged in and charging at a base
DispatchStation	Yes	Driving to a station
ChargeQueueing	No	Waiting at a station for a free stall
ChargingStation	No	Plugged in and charging at a station
DispatchTrip	Yes	Driving to pick up a request
ServicingTrip	Yes	Delivering a request to its destination
DispatchPoolingTrip	Yes	Driving to pick up the first request in a pooled trip
ServicingPoolingTrip	Yes	Traveling along a pickup/drop-off sequence of a pooled trip
OutOfService	No	Vehicle out of fuel and unable to move

**Table 3. Vehicle States**

Most vehicle state transitions are triggered when the fleet control system submits an instruction to a vehicle (see Section 2.3.2), and are referred to as *active* state transitions. For example, if a `DispatchTripInstruction` is sent to an idle autonomous vehicle (and, trivially, on shift), then it will immediately execute that transition. The interface between dispatcher and vehicle is further described in Section 2.3.

Some vehicle state transitions occur without external influence and are referred to as *passive* state transitions. For example, if a vehicle in a DispatchTrip state reaches its target request and that request is still there, then the vehicle will automatically transition into a ServicingTrip state with passengers aboard. Another example is the OutOfService state, which can occur when any agent reaches a fuel level of zero. This is possible due to time spent in any movement state, as well as idle, which can have an idling fuel cost (configurable).

A diagram of the vehicle states and the primary set of allowed state transitions is shown in Figure 4, with detail showing the pathways due to active and passive transitions. In some cases, additional transitions are allowed. If a vehicle state is interruptible, it is possible for the dispatcher to pass new instructions to override that behavior. If a vehicle runs out of energy during the simulation, it is transitioned to OutOfService, regardless of the previous vehicle state.

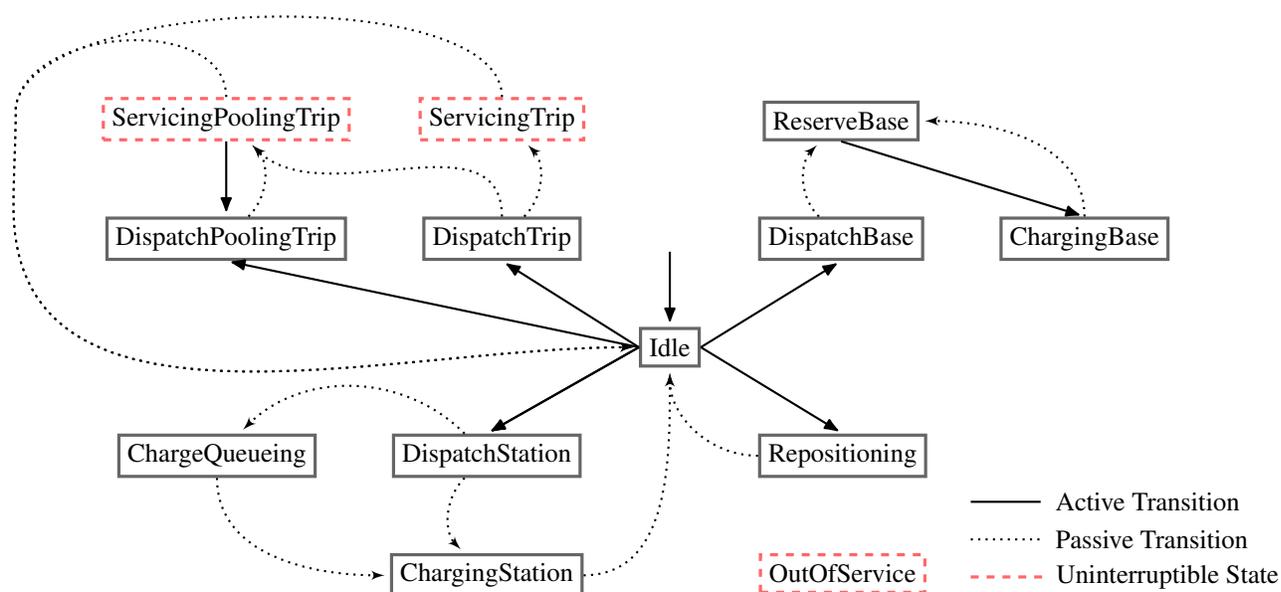


Figure 4. Finite-state machine of vehicle states in HIVE

### 2.2.2 Request

A request is an instance of a mobility-on-demand trip that appears at an origin entity position for some departure time. A request will also have a destination entity position and a number of passengers. All requests will cancel after some globally set duration if they are not serviced. Each request may have a set of unique *memberships* (see Section 2.2.5), but if not provided, any vehicle may service the request. If the request allows pooling, the passengers may enter into a pooling trip with other requests. The complete set of attributes of a request is listed in Table 4.

A globally defined rate structure determines how the price of a request is determined. This is based on a base price along with a price per unit distance, which must meet at least some minimum price.

### 2.2.3 Station

A station represents a location where vehicles can go to refuel. A station can have one or more chargers of variable energy type and energy addition rate. For example, a single station could have five electric chargers with a power of 50 kW and another 10 electric chargers with a power of 7.2 kW. The station locations, as well as the charger types and quantities, are specified as a user input. The complete set of attributes for a station is listed in Table 5.

Attribute	Description	Default
id	Unique identifier	
origin_link	Graph location where request is waiting	
destination_link	Graph location where request is headed	
departure_time	Simulation time the request is activated	
passengers	Number of passengers associated with request	
membership	Set of memberships	∅
price	Price request is paying for service	0.00

**Table 4. Attributes of a Request**

Attribute	Description	Default
id	Unique identifier	
link	Graph location where station is sited	
total_chargers	A map of charger ID to number of total chargers of that type	
available_charger	A map of charger ID to number of available chargers of that type	
on_shift_access_chargers	Set of charger IDs that are accessible to vehicles “on-shift”	
charger_prices_per_kwh	Price of energy for each charger ID (dollars)	0.00
enqueued_vehicles	A map of charger ID to the number of vehicles queued for that charger type	
membership	Set of memberships	∅

**Table 5. Attributes of a Station**

#### 2.2.4 Base

A base represents a location where vehicles can park while not actively seeking out requests. This may represent a fleet depot with several parking spaces and chargers at each space, or a personal garage with a single charger. It may also represent a home location for a human driver where they may have their own private charging capabilities. A base can have an optional station sited at the same location and parked vehicles can access all chargers associated with that station. The complete set of attributes for a base is listed in Table 6.

Attribute	Description	Default
id	Unique identifier	
link	Graph location where base is sited	
total_stalls	The number of total parking spaces	
available_stalls	The number of available parking spaces	
station_id	An optional identifier for an associated station	
membership	Set of memberships	∅

**Table 6. Attributes of a Base**

#### 2.2.5 Membership

In order to enable the modeling of multiple fleets and distinguish between private and public entities (such as a home charger or a public charger), HIVE entities may belong to zero or more memberships. Each entity in the simulation is assigned a membership object that contains a set of membership IDs. The default membership for all entities is an empty set that represents a public membership. During any interaction between two distinct entities (e.g., plugging into a charger or picking up a request) the simulation then compares membership sets to enforce that the interaction is legal.

Public memberships will grant access to any requesting entity, regardless of the membership set of the requesting entity. Private memberships will only grant access to those entities that share at least one membership ID. For example, in order to represent a one-to-one relationship between a human driver and a home charger, each entity would

be assigned a shared membership ID unique to those two entities. Any other vehicle in the simulation would then be restricted from using the home charger due to not holding the unique membership ID.

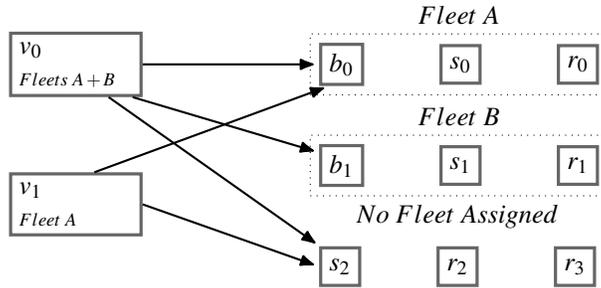


Figure 5. Example of memberships in HIVE

Memberships allow for the modeling of multiple fleets in which entities could belong to one or more fleets, as shown in Figure 5. In the example, vehicle  $v_0$  has a membership to Fleet A and Fleet B and has access to the relevant infrastructure and requests for both fleets. Vehicle  $v_1$ , however, only has a membership to Fleet A, and therefore is restricted from utilizing infrastructure or picking up requests belonging to Fleet B. Both vehicles have access to entities that are not associated with any fleet.

### 2.3 Execution-Level Modeling

The highest level of modeling in HIVE consists of a collection of entities interacting with each other over a set of discrete time steps. At each slice in time, the state of the model is represented with an immutable data structure referred to as the simulation state, which holds the current data for all entities in the model as a snapshot in time. In order to influence the state, a control module must dispatch instructions that capture intent to modify the simulation. This separation of concerns guarantees that the physical and asset-level model behaviors are influenced only by code internal to HIVE, and that user-developed control module extensions do not affect the correctness of the underlying HIVE simulation. All of this occurs within an execution context, which is responsible for “turning the crank” to advance the simulation between discrete simulation time steps.

The overall simulation update process between some simulation state,  $S_t$ , and its successor,  $S_{t+1}$ , is shown in Figure 6. Each phase is explained in the following sections, where the interaction between the simulation state, control module, and execution context is described.

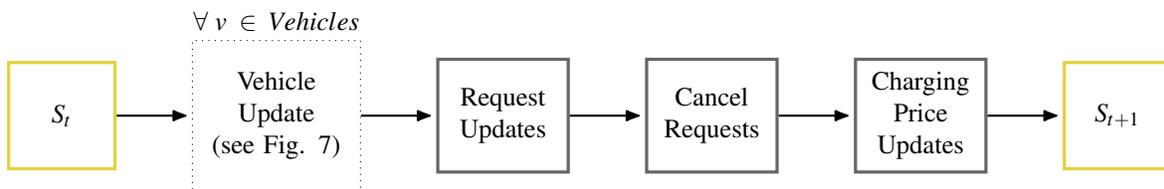


Figure 6. HIVE update process

#### 2.3.1 Simulation State

All of the entities in HIVE (vehicles, requests, stations, and bases) are stored in the simulation state data structure, which maintains simulation-level state information and supports quick entity access by indexing entity IDs and geospatial IDs. Modification of the state is restricted to the internal simulation functions, which govern allowable state transitions and expose those transitions via the instructions abstraction.

Any instance of a simulation state corresponds to a specific, discrete time between simulation events. It contains an instance of each active entity in the simulation at that time, using a collection of indices that support various lookup

operations. Each entity in the simulation is stored in the simulation state within a Hash Array Mapped Trie (HAMT) that supports lookup by the entity ID. This data structure was selected to support the immutability of the simulation state while still providing fast random access.

The simulation state also maintains two HAMT collections of location-based indices. The location index maps an H3 index to an entity ID at a precise resolution, by default using an approximately square-meter hex grid. This index allows the simulation to quickly assess which entities are in the same location. The search index maps an H3 index to an entity ID at a coarser resolution than the first index. This index allows for efficient searching of entities that are nearby to a given entity location due to a property of the H3 index system that allows quick traversal between levels of resolution. By default, this is set at a resolution that produces hex grids that are approximately 5 km<sup>2</sup>.

### 2.3.2 Control Module

The problem of controlling a fleet in HIVE is typically a multi-objective optimization between fleet and driver utility. In HIVE, the interface for all control is the instruction, which is issued by either a fleet dispatcher with a global perspective or a driver with a local perspective.

#### Instructions

The instruction represents the *intent* to modify the state of the system by transitioning a specific vehicle to a new vehicle state, as depicted in Figure 7. Each active transition shown in Figure 4 has a corresponding instruction. Instructions are generated by the control module at each time step, and are interpreted by an internal update mechanism that validates and processes instructions and “steps” the simulation one discrete time step into the future.

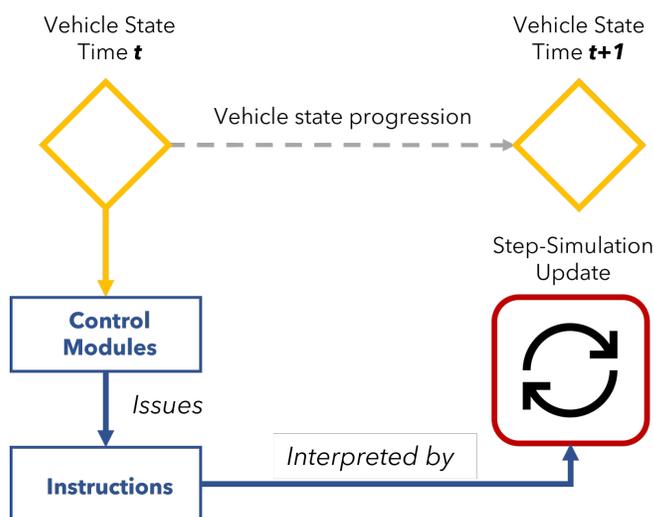


Figure 7. Vehicle update

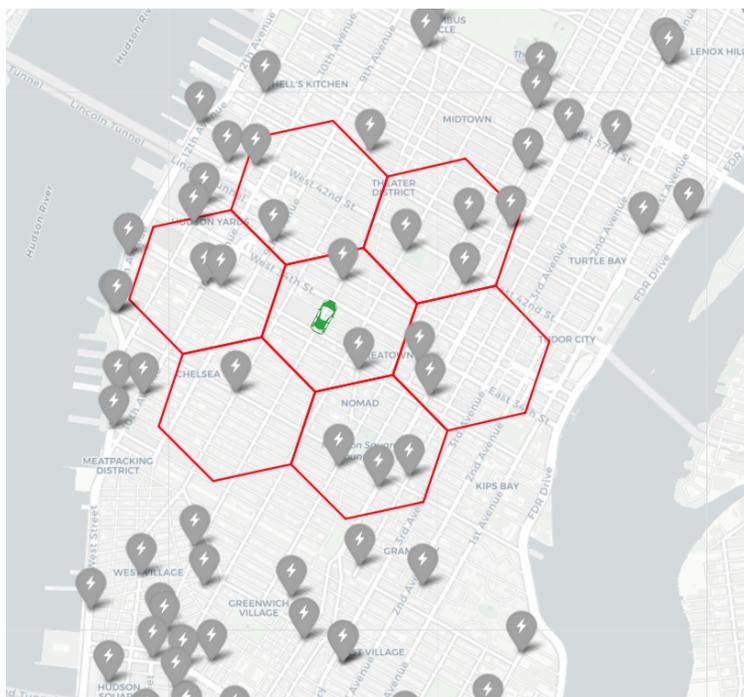
Instructions are validated before they are applied to vehicles; the transition has to meet the preconditions of exiting the previous state as well as any requirements for entering the new state. In this way, the control module does not have a direct means of interfering with the simulation physics, which protects from invalid or illegal transitions. For example, no instruction can be applied to a ServicingTrip Vehicle, since any transition would leave the on-board passengers stranded. Additionally, only one instruction can be issued to a vehicle at each time step of HIVE, and any driver-level instructions can always override any dispatch-level instructions.

#### Dispatcher-Level Control

The dispatcher module is responsible for generating instructions for the vehicle fleet. The design of HIVE allows for users to contribute their own control modules to solve the fleet-level control problem.

The default dispatcher module in HIVE is a two-phase algorithm for charging and matching dispatch. The charging phase conducts a greedy station search for vehicles with less than 20% energy storage. This search leverages the hierarchical nature of the H3 index for fast tracking of position at higher levels of aggregation, as shown in Figure 8. The matching phase solves the request-to-vehicle matching problem by way of the Kuhn-Munkres algorithm [26, 27], using the H3 cell distance between vehicle and request as the cost function. A tiebreaker rule prefers matching instructions when a vehicle is instructed to both charge and match—only the matching instruction is delivered to the vehicle.

The default matching dispatcher does not solve the pooling assignment. Although the underlying simulation physics for pooling exist in HIVE, the integration of a baseline pooling algorithm is left for future work.



**Figure 8. Example of using H3 for a charging station search. The vehicle is seeking a charging location at one of a set of station alternatives, shown as drop points on the map. The search is conducted over an expanding ring of H3 hex grid cells at a higher level of aggregation.**

#### *Driver-Level Control*

In addition to instructions generated at the fleet level, the driver of a vehicle can generate instructions that override any fleet-level instructions. This design is useful for distinguishing between fleet-level and vehicle-level control. The behaviors associated with each driver state (autonomous, human available, and human unavailable) are described below.

The autonomous driver will monitor its time spent idling, and if it exceeds a configurable threshold, the driver will instruct the vehicle to drive to the nearest base location. The vehicle will either charge or transition to a reserve state where the vehicle becomes inactive. In addition, the autonomous driver monitors energy levels during a charge event and instructs the vehicle to stop charging if the state of charge exceeds a configurable threshold.

The human-available driver will monitor its time spent idling and instruct the vehicle to reposition if the time exceeds a configurable threshold. The reposition location is determined as the centroid of the search-level index that has the highest density of active requests. The human-available driver will monitor a charge event in a similar fashion as the autonomous driver.

When a driver enters a human-unavailable driver state, it has reached the end of its scheduled work time and will instruct the vehicle to return to the home base location, assuming it is in an interruptible vehicle state. The driver will decide if additional fueling is required in order to guarantee that the vehicle energy exceeds a start-of-shift threshold (configurable) prior to beginning service on the following day. The vehicle may be fueled on the journey to the home location or plugged into a possible home charger in order to meet this threshold.

An example of the process for updating vehicles is shown in Figure 9. In this example, the default HIVE control module is used, and one single update event (at time  $t$ ) for a single vehicle is considered. The vehicle begins in the idle state and is examined by the charging dispatcher. The dispatcher determines that the vehicle needs to charge and issues a charge instruction. Next, the matching dispatcher determines that this vehicle matches the request in the system and issues a dispatch trip instruction. Finally, the driver control module, with full knowledge of the vehicle state and the two instructions that have already been generated, decides to not issue a final instruction. The instruction that was issued last in the pipeline of controllers is popped from the instruction stack and applied to the vehicle. In this case, the dispatch trip instruction takes priority over the charge instruction and the vehicle transitions into the dispatch trip state.

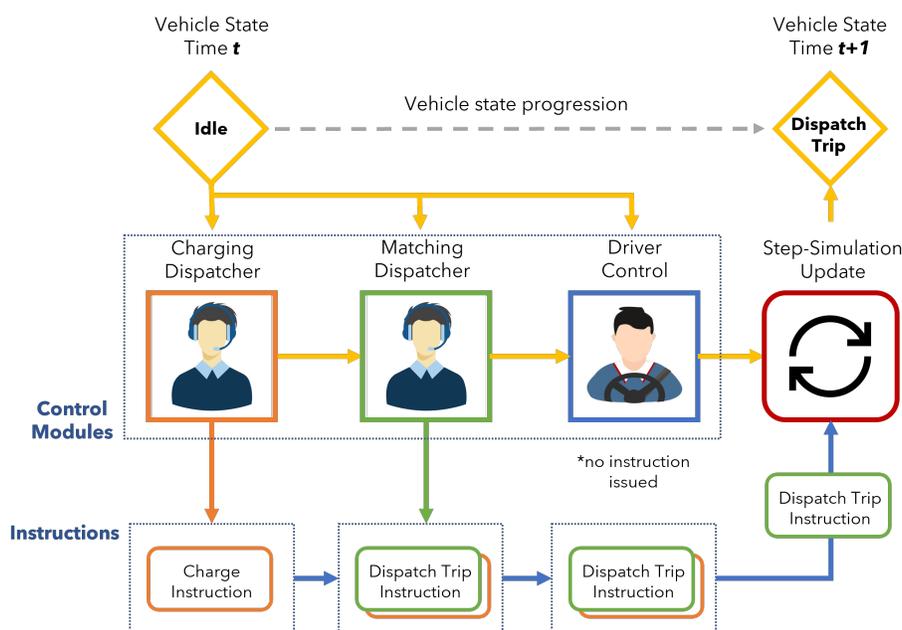


Figure 9. Example vehicle state update

### 2.3.3 Execution Context

It is only when HIVE is run in an execution context that the simulation is advanced beyond the current snapshot captured in the simulation state. HIVE will execute this in a fixed-increment, discrete event simulation governed by a series of update functions, which can modify different aspects of HIVE. Updates occur via a fixed-increment time progression versus next-event time progression, as every vehicle requires a state update in every time interval. Running a complete day of HIVE is simply the repeated application of update functions until the configured stop time. The following sections describe time stepping and instantaneous simulation updates.

#### Time Stepping Update

Advancing time in HIVE involves calling a step method on all state machines (vehicle and driver). By default, if no instructions are issued, a step will continue the current activity governed by the vehicle state, unless overridden by the driver state or if a terminal state is reached, where a passive transition will be applied. In the case that an instruction is issued by the control module, it will be applied, which may fail. Instructions are not trusted; for example, a

user-created dispatcher module may incorrectly dispatch a vehicle to a location outside of the road network. If an instruction such as this fails to apply, the vehicle remains with its default step behavior.

### Instantaneous Updates

Included in HIVE are three default update functions that manage updating state to resources based on the time of day.

The requests update function reads in a file stream of requests and adds them to the simulation if the origin time falls within the window of  $[t - 1, t]$ .

The cancel requests update function removes any requests that have expired, meaning their origin time plus a request cancel time delta (configurable) is less than the current simulation time.

The charging price update function updates the cost of energy throughout the simulation. The cost of energy is provided as step function of cost (dollars/kWh) over time. A unique step function can be specified for each station, or a unique step function can be specified over an entire H3 hexagon of any resolution.

## 2.4 Model Inputs

There are a variety of file dependencies in order to run a HIVE simulation, as shown in Figure 10. Each entity file type lists unique identifiers and attributes of the various entities in a simulation. The optional fleets file assigns the membership relations between entities. VehicleTypes and schedules extend the capabilities of vehicles with corresponding ID assignments. Chargers lists the types of chargers in a simulation that can be referenced by a station. Additional files set the behaviors for the supply-side modeling.

Some files and fields within files are purely optional. For example, omitting a station ID for a base indicates that there is no on-base charging infrastructure. Independent files are referenced in a top-level scenario input file, which can be used to create a suite of test cases and model versioning.

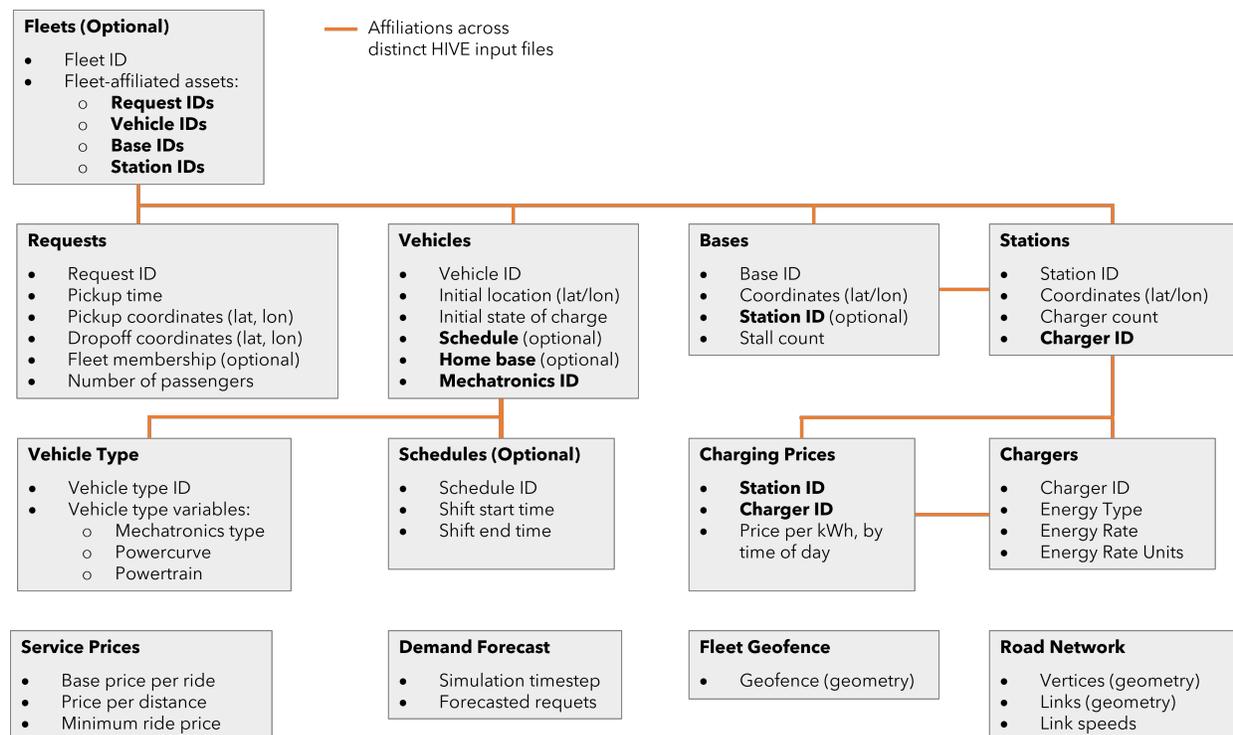


Figure 10. Inputs to the HIVE model

## 3 Case Study

### 3.1 Case Study Inputs

This section introduces a sample application of HIVE, which is intended to demonstrate many of the features of the model. The scenario discussed is available as an in-built example and comes prepackaged in the model repository. Note that the purpose of this section is illustrative and not intended to closely depict real-world inputs or convey research findings. Descriptions of key input variables follow.

- The case study models a fleet of 1,000 human-driven vehicles across two fleets—250 Yellow Cab Taxis (YCB) and 750 for-hire vehicles (FHVs)—responding to requests in Manhattan.
- Trip requests for each fleet segment are sourced from the New York Taxi Cab and Limousine Commission [28]. In total, 20,000 YCB requests and 15,000 FHV requests are sourced from March 8, 2019, a day of significant ride-hailing activity. The vehicles and requests selected produce an intentionally undersized YCB fleet (80 requests per vehicle) and an adequately sized FHV fleet (20 requests per vehicle). Requests are exclusively available to the corresponding fleet segment, as enforced through the membership field.
- Driver shift times are synthesized from shift distributions provided by the Taxi Cab and Limousine Commission [29]. Shift end times are calculated by assuming shift lengths of between 6–8 hours for FHVs and 10–14 hours for YCBs, reflecting observed differences in typical hours of operation across segments. These variables contribute to dynamic fleet sizing throughout the day, as individual drivers transition between being on- and off-shift.
- Vehicle home locations are sampled randomly throughout the fleet geofence within Manhattan.
- Eighty-five 50-kW charging plugs are sited across 10 stations throughout Manhattan. These stations—containing either 5 or 15 plugs—are also modeled as either exclusive to a single fleet or shared across all vehicles. No overnight charging access is provided to vehicles.
- Vehicles are modeled as sedans with 62.5-kWh batteries, 50-kW peak charge acceptance, and an efficiency of 250 Wh/mi.
- Finally, the fleet is initialized with each vehicle starting at its home location with state-of-charge values sampled randomly from 30% to 70%.

### 3.2 Results

The following discussion of results illustrates sample outcomes from the simulation, but represents only a portion of available insights from simulation logs. HIVE's flexible reporting enables logging results across all assets at each time step, providing insight into results at the vehicle, asset, and fleet levels. Figure 11 provides an overview of sample information available for each vehicle during the simulation. The figure illustrates the full activity of a sample YCB vehicle modeled as being on-shift from approximately 7 a.m. to 7:30 p.m. servicing trip requests throughout Manhattan. The vehicle trajectory contains a variety of different driving states including commuting, repositioning, dispatching to trips, performing trips, and dispatching to charge. During each driving event, the vehicle is required to traverse the roadway network with travel times determined by road link speeds. Although links in Manhattan are low-speed, the vehicle accumulates significant mileage throughout the day—over 150 miles—requiring a fast charge at a station in midtown Manhattan before continuing to service trips throughout the remainder of its shift. Upon completing its shift, the vehicle completes its final ride to its home location in uptown Manhattan.

In addition to vehicle information, HIVE files also contain a summary of charging station data. Figure 12 illustrates load profiles corresponding to three sample stations sited and used by the fleet during the day simulated. The stations selected vary by fleet membership affiliation, including an FHV exclusive station, a YCB exclusive station, and a shared station. Although only a minority of station loads are shown, inspection of the results from these three stations reveals several interesting insights. First, the majority of the charging load is attributed to the YCB fleet despite it only comprising 25% of fleet vehicles. This is due to the larger number of requests exposed to the YCB fleet and the longer shift lengths assumed versus the FHV fleet. Additionally, the FHV and YCB fleets appear to

### Sample Vehicle Day

- Shift start time: 6:58 AM
- Shift end time: 7:24 PM
- Shift length: 12h, 26m
- Fleet: Yellow cab

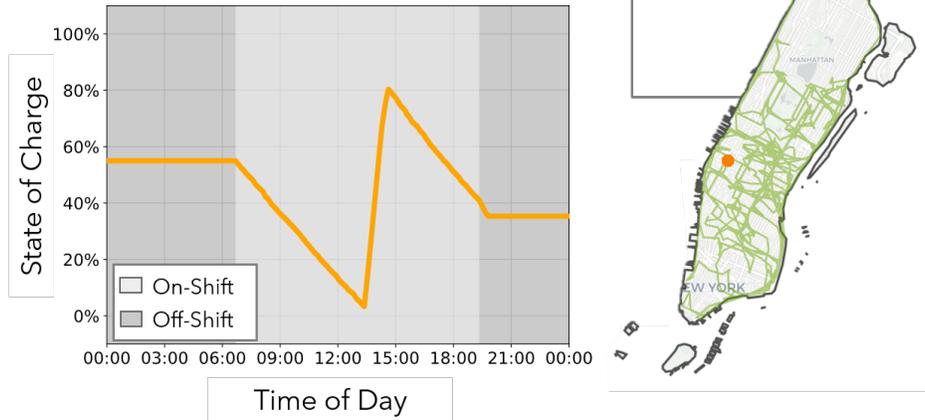


Figure 11. Sample vehicle trajectory and state-of-charge evolution throughout a 24-hour simulation

require charging at similar times, resulting in competition for plugs at shared stations. The timing of charge events is predominantly influenced by shift start times and request dynamics that induce vehicles to drive and consume battery energy. Finally, the loads across stations are observed to be highly variable throughout the day, with no charging during overnight hours while the majority of the fleet is off-shift. Although not shown, fleetwide load profiles may be trivially calculated by aggregating the fleet load across all stations and overnight chargers.

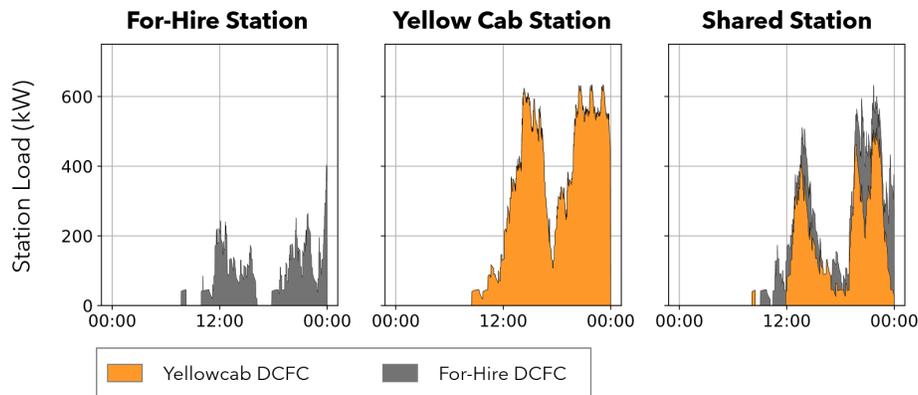
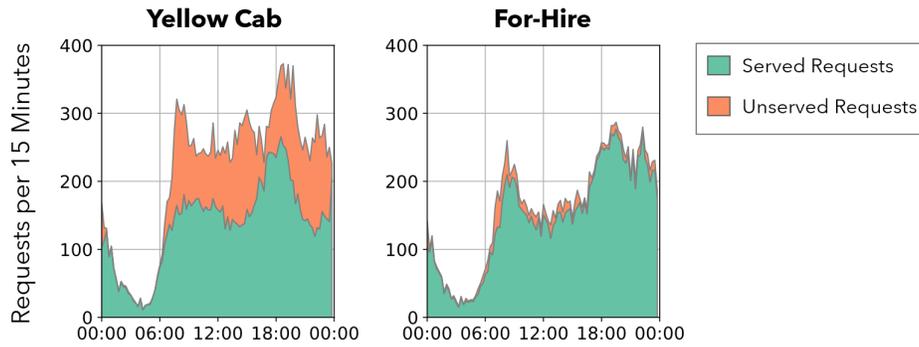


Figure 12. Sample load profiles from stations in HIVE with varying membership access types (FHV exclusive, YCB exclusive, and shared)

Finally, HIVE logs provide insight regarding the performance of the overall fleet. Figure 13 shows the timing of all requests exposed to the fleet segments, categorized as either being served or unserved. Requests are categorized as served if a vehicle is available to transport a passenger and unserved if a vehicle cannot arrive in a specified amount of time after the request appears (10 minutes in this case). Outcomes at the fleet level demonstrate the importance of fleet sizing relative to request quantity. Whereas the FHV fleet effectively serves nearly all requests, the YCB fleet completes only 64.4% of requests due to fewer vehicles available to accommodate passengers.



**Figure 13. Request service across YCB and FHV fleets exposed to distinct request populations**

Although limited in size and geographic scale, the sample use case demonstrates the flexibility of HIVE to model ride-hailing fleets and capture trade-offs between input variables of interest (in this case, the fleet size on load profiles and quality of service). Note that the functionality of HIVE extends well beyond considerations discussed here. Additional opportunities for inquiry are vast, including topics ranging from trade-offs between vehicle battery investment and charging investment, autonomous fleet operation versus human-driven fleets, and more.

## 4 Conclusion and Future Work

This report describes the motivation behind creation of the HIVE simulation tool and model architecture, and demonstrates its functionality with a simple case study. The tool was developed to support internal research programs and to be integrated with existing in-lab platforms. The model architecture is broken out into three distinct levels and was crafted to support future extension points and flexibility of introducing new control strategies. The case study highlights the model usage and some of the insights that can be derived from the high-density outputs. Some immediate areas for future work exist in the areas of co-simulation, high-performance computing, and learning-based control.

While built on simple constructs, HIVE reveals an emergent complexity due to the interaction of many agents in an agent-based model. Within the simulation period, there are opportunities to produce more realistic results by integrating HIVE with other simulation tools “in the loop.” For example, an electric grid model based on real-world grid data could replace the tabular charging prices lookup approach, in order to study the effects of electric MoD load on the power grid. HIVE could also read link speed outputs from traffic simulations in order to evaluate runs using simulated congestion effects. Also, replacing tabular engine models with dynamic co-simulation would produce more realistic energy consumption.

As simulation sizes scale, agent-based models like HIVE are subject to a nonlinear increase in complexity proportional to the density of active agents. Additionally, certain aspects of HIVE such as matching dispatch rely on matrix-based representations, which become intractable when simulating many agents. One can observe in the modeling of MoD services that an intuitive means for dividing into sub-problems falls along the lines of dispatch regions. By applying geofencing, the problem can be subdivided into a set of distributed dispatchers, all covering separate, disjointed, and contiguous regions of the study area. The ability to reformulate HIVE as a distributed simulation composed of geofenced HIVE instances comes “for free” as a result of making these assumptions. The speed of a HIVE simulation can then be improved, or HIVE can then be run for arbitrarily large scenarios or at a finer level of modeling detail. For example, exploring regional-scale scenarios with very large, heterogeneous fleets becomes feasible.

Finally, data-driven control policies for fleets of MoDs is an emerging area of active research. In many simulation models, the non-stationarity of the state space due to traffic simulation poses a great challenge to optimal policy learning. Such effects are not present in the HIVE model. Although this is a shortcoming for certain research problems, it can be an acceptable assumption when learning strategies for fleet management. Additionally, the learned policy may be transferred after training to the more complex model. In particular, the driver control module, along with the repositioning and charging assignment fleet control modules, are suitable applications for both centralized and multi-agent deep reinforcement learning algorithms.

## References

- [1] H. Ma, F. Balthasar, N. Tait, X. Riera-Palou, and A. Harrison. “A new comparison between the life cycle greenhouse gas emissions of battery electric vehicles and internal combustion vehicles.” *Energy Policy* 44 (2012), pp. 160–173. ISSN: 0301-4215. DOI: <https://doi.org/10.1016/j.enpol.2012.01.034>. <https://www.sciencedirect.com/science/article/pii/S0301421512000602>.
- [2] Q. Wang, M. A. DeLuchi, and D. Sperling. “Emission Impacts of Electric Vehicles.” *Journal of the Air & Waste Management Association* 40.9 (1990), pp. 1275–1284. DOI: [10.1080/10473289.1990.10466782](https://doi.org/10.1080/10473289.1990.10466782). eprint: <https://doi.org/10.1080/10473289.1990.10466782>. <https://doi.org/10.1080/10473289.1990.10466782>.
- [3] P. Weldon, P. Morrissey, and M. O’Mahony. “Long-term cost of ownership comparative analysis between electric vehicles and internal combustion engine vehicles.” *Sustainable Cities and Society* 39 (2018), pp. 578–591. ISSN: 2210-6707. DOI: <https://doi.org/10.1016/j.scs.2018.02.024>. <https://www.sciencedirect.com/science/article/pii/S2210670717314403>.
- [4] E. W. Wood, C. L. Rames, M. Muratori, S. Srinivasa Raghavan, and M. W. Melaina. *National plug-in electric vehicle infrastructure analysis*. Tech. rep. National Renewable Energy Lab.(NREL), Golden, CO (United States), 2017.
- [5] T. S. Combs, L. S. Sandt, M. P. Clamann, and N. C. McDonald. “Automated Vehicles and Pedestrian Safety: Exploring the Promise and Limits of Pedestrian Detection.” *American Journal of Preventive Medicine* 56.1 (2019), pp. 1–7. ISSN: 0749-3797. DOI: <https://doi.org/10.1016/j.amepre.2018.06.024>. <https://www.sciencedirect.com/science/article/pii/S0749379718320932>.
- [6] M. Taiebat, A. L. Brown, H. R. Safford, S. Qu, and M. Xu. “A review on energy, environmental, and sustainability implications of connected and automated vehicles.” *Environmental science & technology* 52.20 (2018), pp. 11449–11465.
- [7] J. W. Ward, J. J. Michalek, C. Samaras, I. L. Azevedo, A. Henao, C. Rames, and T. Wenzel. “The impact of Uber and Lyft on vehicle ownership, fuel economy, and transit across U.S. cities.” *iScience* 24.1 (2021), p. 101933. ISSN: 2589-0042. DOI: <https://doi.org/10.1016/j.isci.2020.101933>. <https://www.sciencedirect.com/science/article/pii/S2589004220311305>.
- [8] A. Henao and W. E. Marshall. “The impact of ride-hailing on vehicle miles traveled.” *Transportation* 46.6 (Dec. 1, 2019), pp. 2173–2194. ISSN: 1572-9435. DOI: [10.1007/s11116-018-9923-2](https://doi.org/10.1007/s11116-018-9923-2). <https://doi.org/10.1007/s11116-018-9923-2> (visited on 06/02/2021).
- [9] P. M. Boesch and F. Ciari. “Agent-based simulation of autonomous cars.” *2015 American Control Conference (ACC)*. 2015, pp. 2588–2592. DOI: [10.1109/ACC.2015.7171123](https://doi.org/10.1109/ACC.2015.7171123).
- [10] M. Maciejewski, J. M. Salanova, J. Bischoff, and M. Estrada. “Large-scale microscopic simulation of taxi services. Berlin and Barcelona case studies.” *Journal of Ambient Intelligence and Humanized Computing* 7.3 (June 1, 2016), pp. 385–393. ISSN: 1868-5145. DOI: [10.1007/s12652-016-0366-3](https://doi.org/10.1007/s12652-016-0366-3). <https://doi.org/10.1007/s12652-016-0366-3> (visited on 02/08/2020).
- [11] D. J. Fagnant and K. M. Kockelman. “The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios.” *Transportation Research Part C: Emerging Technologies* 40 (Mar. 1, 2014), pp. 1–13. ISSN: 0968-090X. DOI: [10.1016/j.trc.2013.12.001](https://doi.org/10.1016/j.trc.2013.12.001). <http://www.sciencedirect.com/science/article/pii/S0968090X13002581> (visited on 02/08/2020).
- [12] J. Zachariah, J. Gao, A. Kornhauser, and T. Mufti. “Uncongested Mobility for All: A Proposal for an Area Wide Autonomous Taxi System in New Jersey.” Transportation Research Board 93rd Annual Meeting Transportation Research Board. 2014. <https://trid.trb.org/view/1288288> (visited on 02/09/2020).
- [13] H. Durrant-Whyte, N. Roy, and P. Abbeel. “Load Balancing for Mobility-on-Demand Systems.” *Robotics: Science and Systems VII*. 2012, pp. 249–256.
- [14] T. D. Chen, K. M. Kockelman, and J. P. Hanna. “Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle & charging infrastructure decisions.” *Transportation Research Part A: Policy and Practice* 94 (Dec. 1, 2016), pp. 243–254. ISSN: 0965-8564. DOI: [10.1016/j.tra.2016.08.020](https://doi.org/10.1016/j.tra.2016.08.020). <http://www.sciencedirect.com/science/article/pii/S096585641630756X> (visited on 02/08/2020).
- [15] J. Bischoff and M. Maciejewski. “Agent-based Simulation of Electric Taxicab Fleets.” *Transportation Research Procedia*. Sustainable Mobility in Metropolitan Regions. mobil.TUM 2014. International Scien-

- tific Conference on Mobility and Transport. Conference Proceedings. 4 (Jan. 1, 2014), pp. 191–198. ISSN: 2352-1465. DOI: [10.1016/j.trpro.2014.11.015](https://doi.org/10.1016/j.trpro.2014.11.015). <http://www.sciencedirect.com/science/article/pii/S2352146514002981> (visited on 02/08/2020).
- [16] B. Loeb, K. M. Kockelman, and J. Liu. “Shared autonomous electric vehicle (SAEV) operations across the Austin, Texas network with charging infrastructure decisions.” *Transportation Research Part C: Emerging Technologies* 89 (Apr. 1, 2018), pp. 222–233. ISSN: 0968-090X. DOI: [10.1016/j.trc.2018.01.019](https://doi.org/10.1016/j.trc.2018.01.019). <http://www.sciencedirect.com/science/article/pii/S0968090X18300627> (visited on 02/08/2020).
- [17] P. M. Bösch, F. Becker, H. Becker, and K. W. Axhausen. “Cost-based analysis of autonomous mobility services.” *Transport Policy* 64 (May 1, 2018), pp. 76–91. ISSN: 0967-070X. DOI: [10.1016/j.tranpol.2017.09.005](https://doi.org/10.1016/j.tranpol.2017.09.005). <http://www.sciencedirect.com/science/article/pii/S0967070X17300811> (visited on 01/30/2020).
- [18] H. Zhang, C. J. R. Sheppard, T. E. Lipman, T. Zeng, and S. J. Moura. “Charging infrastructure demands of shared-use autonomous electric vehicles in urban areas.” *Transportation Research Part D: Transport and Environment* 78 (Jan. 1, 2020), p. 102210. ISSN: 1361-9209. DOI: [10.1016/j.trd.2019.102210](https://doi.org/10.1016/j.trd.2019.102210). <https://www.sciencedirect.com/science/article/pii/S1361920919303840> (visited on 07/08/2021).
- [19] G. S. Bauer, A. Phadke, J. B. Greenblatt, and D. Rajagopal. “Electrifying urban ridesourcing fleets at no added cost through efficient use of charging infrastructure.” *Transportation Research Part C: Emerging Technologies* 105 (Aug. 1, 2019), pp. 385–404. ISSN: 0968-090X. DOI: [10.1016/j.trc.2019.05.041](https://doi.org/10.1016/j.trc.2019.05.041). <http://www.sciencedirect.com/science/article/pii/S0968090X19300713> (visited on 01/30/2020).
- [20] K. M. Gurumurthy. “Shared autonomous vehicle system designs for major metro areas : an examination of geofencing, ride-sharing, stop-location, and drivetrain decisions.” Thesis. Dec. 4, 2020. DOI: [10.26153/tsw/12409](https://doi.org/10.26153/tsw/12409). <https://repositories.lib.utexas.edu/handle/2152/85445> (visited on 07/08/2021).
- [21] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. *OpenAI Gym*. 2016. eprint: [arXiv:1606.01540](https://arxiv.org/abs/1606.01540).
- [22] E. Ucer, I. Koyuncu, M. C. Kisacikoglu, M. Yavuz, A. Meintz, and C. Rames. “Modeling and Analysis of a Fast Charging Station and Evaluation of Service Quality for Electric Vehicles.” *IEEE Transactions on Transportation Electrification* 5.1 (Mar. 2019), pp. 215–225. ISSN: 2332-7782. DOI: [10.1109/TTE.2019.2897088](https://doi.org/10.1109/TTE.2019.2897088).
- [23] G. J. W. L. W. E. e. a. Brooker A. “FASTSim: A Model To Estimate Vehicle Efficiency, Cost, and Performance.” *SAE Technical Paper* (2015).
- [24] I. Brodsky. “H3: Uber’s hexagonal hierarchical spatial index.” Available from Uber Engineering website: [https://eng.uber.com/h3/\[22 June 2019\]](https://eng.uber.com/h3/[22 June 2019]) (2018).
- [25] G. Boeing. “OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks.” *Computers, Environment and Urban Systems* 65 (2017), pp. 126–139.
- [26] H. W. Kuhn. “The Hungarian method for the assignment problem.” *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.
- [27] J. Munkres. “Algorithms for the assignment and transportation problems.” *Journal of the society for industrial and applied mathematics* 5.1 (1957), pp. 32–38.
- [28] T. A. L. Commission. *TLC Trip Record Data*. <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page> (visited on 07/15/2021).
- [29] Taxi & Limousine Commission. *2018 TLC Factbook*. New York City: NYC TLC, Sept. 2018. [https://www1.nyc.gov/assets/tlc/downloads/pdf/2018\\_tlc\\_factbook.pdf](https://www1.nyc.gov/assets/tlc/downloads/pdf/2018_tlc_factbook.pdf).